

BWS-1000 Z80 MONITOR PROGRAM
USER'S MANUAL

ByteWyde Systems

**P.O. Box 146
CANTERBURY
Victoria 3126**

BWS-1000 Z80 Monitor Program

BWS-1000 Z80 M O N I T O R P R O G R A M

USER'S MANUAL

December 1983

July 1984

January 1985

Monitor Version 3.23

Manual Rev 1.3

Copyright (C) 1983, 1984, 1985

ByteWyde Systems

P.O. Box 146

CANTERBURY

Victoria 3126

Rev 1.3

CONTENTS		Page

1.0	Introduction	1.1
	1.1 General Description	
	1.2 Operation	
2.0	Hardware Requirements and Configuration	2.1
	2.1 Prolog 7806 processor card	
	2.2 Prolog 7803 processor card	
	2.3 Prolog 7804 processor card	
	2.4 Prolog 7880 processor card	
	2.5 Prolog 7304 dual UART card	
3.0	Command Summary and Syntax	3.1
	3.1 Command Summary	
	3.2 Command Syntax	
	3.3 Symbol Definitions	
4.0	Command Descriptions	4.1
	4.1 '?' - Display a command summary	
	4.2 'A' - Set/examine/remove address compare	
	4.3 'B' - Set/examine/remove software breakpoints	
	4.4 'C' - Call Command Basic (if installed)	
	4.5 'D' - Display memory block	
	4.6 'F' - Fill a memory block	
	4.7 'G' - Go to target execution address	
	4.8 'H' - Hexadecimal calculator	
	4.9 'I' - Input data via i/o port	
	4.10 'K' - Copy a memory block	
	4.11 'L' - Load an INTEL-HEX format file	
	4.12 'M' - Examine/alter memory byte	
	4.13 'O' - Output data via i/o port	
	4.14 'P' - Examine/alter/trace register pair	
	4.15 'Q' - Test a memory block	
	4.16 'R' - Examine/alter single register	
	4.17 'S' - Single step target code	
	4.18 'T' - Talk to host CPU	
	4.19 'U' - Upload an INTEL-HEX format file	
	4.20 'V' - Compare (verify) two memory blocks	
	4.21 'W' - Examine/alter/trace memory word	
	4.22 'X' - Display saved CPU state	
	4.23 'Z' - Set/alter breakpoint display flag	
5.0	Monitor Entry Points	5.1
	5.1 Monitor Subroutines	
	5.2 Interrupt Jump Vectors	
Appendix A	Command Syntax Summary	A.1

1.0 INTRODUCTION

1.1 General Description

ByteWyde Systems BWS-1000 Z80 monitor program is a high performance stand alone terminal based monitor utility package suitable for use on any STD bus-based Z80 microprocessing system. It is designed to provide a support environment for the development of STD-based hardware, simple stand alone software, and for the debugging of larger software packages which have been assembled on a host development system. The monitor is particularly suitable to assist with the development of real-time interrupt driven software, such as multi-tasking operating systems or instrumentation software.

A comprehensive range of functions are provided as standard, with further enhanced functions available when ByteWyde Systems BWS-2000 Hardware Debug card is included as an option.

STANDARD FUNCTIONS:

- . examine/alter memory bytes and words
- . display/copy/verify memory blocks
- . fill/clear/test memory blocks
- . input/output bytes through I/O ports
- . display/examine/alter/set CPU registers and register pairs
- . jump to commence program execution
- . set up to 16 simultaneous software breakpoints in user programs (RAM based program only).
- . select breakpoint re-entry display format (one of 4)
- . send/receive memory dumps as INTEL-HEX FORMAT files
- . display help menu, describing functions available
- . talk to host CPU (terminal mode)

ADDITIONAL FUNCTIONS SUPPORTED BY THE BWS-2000 HARDWARE DEBUG CARD

- . single step any user program (ROM or RAM based)
- . set a breakpoint trap on address bus access to any memory location
- . multiple passes through software breakpoints before re-entering the monitor
- . warm boot from "hung" user programs to avoid re-initialising registers, breakpoints, etc.

1.0 Introduction (contd)

The monitor is currently configured to be used with Prolog's new 7806 stand alone Z80 microprocessor card, as well as with the Prolog 7803, 7804, 7880 Z80 microprocessor cards (when combined with either the 7301 or 7304 serial UART cards). Other Z80 cards can be easily accommodated by minor revisions to the monitor BIOS subroutines.

1.2 Operation

The monitor is supplied in a single 2732 or 2764 EPROM. Details of the hardware configuration to be set up for the Z80 card (and UART card if required) are given in this manual which is supplied with the EPROM. No other support software or operating system is required for the monitor to operate, although host systems can communicate with the monitor using the INTEL-HEX file transfer facility.

The monitor program occupies addresses 0000 to 0FFF Hex, and hence will commence execution from a Z80 RESET. All software restart and interrupt jumps are vectored through monitor RAM to allow a user program to access the interrupt structure. The monitor uses the RST 3 software restart for software breakpoints. If the BWS-2000 Hardware Debug card is installed, the monitor uses the NMI interrupt for hardware breakpoint functions.

The monitor recognises the presence of the BWS-2000 Hardware Debug card in a system, so this card may be retro-fitted at any time. When the monitor recognises the Hardware Debug card in the system, it will ring the terminal bell upon a cold boot.

User programs should not use the monitor's stack area, especially if it is desired to use the monitor's software trace facilities. User programs requiring a stack should initialise the stack pointer as necessary. Of course, user programs should not use the monitor RAM area, and expect the monitor to work.

2.0 HARDWARE REQUIREMENTS AND CONFIGURATION

The minimum STD system necessary to run the BWS-1000 Z80 monitor program must include:

- . one STD Z80 processor card, Prolog types 7806, 7803, 7804, 7880, or equivalent.
- . at least one RS-232C serial port. The standard baud rate for communication to the VDU is 9600 baud.
- . ROM address space from 0000 Hex to 0FFF Hex.
- . 200 Hex locations of RAM, located as discussed below.
- . One terminal VDU, with RS-232C serial interface.

If it is desired to use the monitor's INTEL-HEX load and transmit functions, a second RS-232C serial port must be installed into the system. The default baud rate for this port is 2400 baud.

The BWS-1000 Z80 monitor program resides in one Intel 2732 EPROM or equivalent, and occupies addresses 0000 through 0FFF Hex. Program parameters and stack require some 200 Hex locations of RAM, which is usually at the start of a 6116 RAM. The address of the monitor RAM area depends on the particular STD Z80 processor card in use.

The following versions of the BWS-1000 monitor program match the particular Prolog Z80 processor cards listed:

BWS-1000	----	Prolog 7806-0 Z80 processor card
BWS-1001	----	Prolog 7806-1 Z80 processor card
BWS-1002	----	Prolog 7803 Z80 processor card with Prolog 7304 dual UART card
BWS-1003	----	Prolog 7804 and 7880 Z80 processor cards with Prolog 7304 dual UART card
BWS-1009	----	CP/M-80 version

For each of the BWS-1000 Z80 monitor program versions listed above, the specific hardware configuration requirements are discussed below. For unique configurations not covered by this discussion (especially for different baud rates and different memory maps), contact ByteWyde Systems for further details.

- NOTES: 1) Prolog's 7806-0 and 7806-1 Z80 processor cards have slightly different clock rates. This affects the BAUD rate of the RS-232C serial ports.
- 2) Users must be careful to order the correct version of the BWS-1000 Z80 monitor program to suit their particular Prolog Z80 processor card as indicated above.

2.0 Hardware Requirements and Configuration (contd)

2.1 Prolog 7806 Z80 Processor Card - Monitor BWS-1000 & BWS-1001

The 7806 card provides all necessary hardware facilities on one card, i.e. two RS-232C serial ports and four 28 pin JEDEC memory sockets. Links on this card must be set up to the factory installation, with memory mapping option 0, as per the Prolog manual, to provide:

- . Memory socket 0 address mapped to 0000 to 0FFF Hex as 2732 or 2764 type EPROM.
- . Memory socket 1 address mapped to 1000 to 1FFF Hex for user use.
- . Memory socket 2 address mapped to 2000 to 2FFF Hex for user use.
- . Memory socket 3 address mapped to 3000 to 37FF Hex as 6116 type RAM.
- . Serial Port 1 set up as Data Communication Equipment

In addition, Serial Port 2 should be set up as Data Terminal Equipment.

The terminal VDU is connected to Serial port 1, which is initialised to 9600 baud, 8 data bits, no parity bit, 1 stop bit, no interrupts. CTS and DSR are asserted.

RS-232C connections for port 1 are as follows:

pin 1	protective ground
pin 2	data received by 7806 card
pin 3	data transmitted from 7806 card
pin 4	RTS sensed by 7806 card (not used by monitor)
pin 5	CTS asserted by 7806 card
pin 6	DSR asserted by data terminal
pin 7	signal ground
pin 20	DTR sensed by 7806 card (not used by monitor)

Serial port 2 is initialised to 2400 baud, 8 data bits, no parity bits, 1 stop bit, no interrupts. DTR and RTS are asserted.

RS-232C connections for port 2 are as follows:

pin 1	protective ground
pin 2	data transmitted from 7806 card
pin 3	data received from 7806 card
pin 4	RTS asserted by 7806 card
pin 5	CTS sensed by 7806 card (not used by monitor)
pin 6	DSR sensed by 7806 card (not used by monitor)
pin 7	signal ground
pin 20	DTR asserted by 7806 card

NOTE: Consult the 7806 processor manual to convert serial port 2 to Data Communication Equipment. The monitor is unaffected.

2.0 Hardware Requirements and Configuration (contd)

To install the BWS-1000 Z80 monitor program insert EPROM BWS-1000 or BWS-1001 as appropriate into memory socket 0 and a 6116 RAM chip into memory socket 3. Set the 7806 card links as specified and insert it into an STD card frame. Connect the terminal, turn on the power, press pushbutton reset, and the monitor should display a sign-on prompt on the VDU terminal. No other initialisation is required.

2.2 Prolog 7803 Z80 Processor Card - Monitor BWS-1002

The 7803 card requires a separate dual UART card to support the BWS-1000 Z80 monitor program. The standard monitor version (BWS-1002) assumes the Prolog 7304 dual UART card is installed in the system. Setup requirements for this UART card are discussed in section 2.5.

Links on the 7803 card itself must be set up to the factory installation as per the Prolog manual, to provide:

- . Memory socket 0 address mapped to 0000 to 07FF Hex as 2716 type EPROM.
- . Memory socket 1 address mapped to 0800 to 0FFF Hex as 2716 type EPROM.
- . Memory socket 2 address mapped to 1000 to 17FF Hex for user use.
- . Memory socket 3 address mapped to 1800 to 1FFF Hex for user use.
- . RAM memory sockets mapped to 2000 Hex to 2FFF Hex

To install the BWS-1000 Z80 monitor program, insert the two EPROMs comprising the BWS-1002 monitor program into memory sockets 0 and 1, and 2114 RAM chips into the memory RAM sockets. Set the 7803 card links as specified and insert it into an STD card frame. Set the 7304 card links as described in section 2.4, and insert it into the STD card frame. Connect the terminal, turn on the power, press pushbutton reset, and the monitor should display a sign-on prompt on the VDU terminal. No other initialisation is required.

BWS-1000 Z80 Monitor Program

2.0 Hardware Requirements and Configuration (contd)

2.3 Prolog 7804 Z80 Processor Card - Monitor BWS-1003

The 7804 card requires a separate dual UART card to support the BWS-1000 Z80 monitor program. The standard monitor version (BWS-1003) assumes that the Prolog 7304 dual UART card is installed in the system. Setup requirements for this UART card are discussed in section 2.5.

Links on the 7804 card must be set up to the factory installation as per the Prolog manual, to provide:

- . Memory socket 0 address mapped to 0000 to 0FFF Hex as 2732 type EPROM.
- . Memory socket 1 address mapped to 1000 to 1FFF Hex for user use.
- . Memory socket 2 address mapped to 2000 to 2FFF Hex for user use.
- . Memory socket 3 address mapped to 3000 to 37FF Hex as 6116 type RAM.

To install the BWS-1000 Z80 monitor program, insert EPROM BWS-1003 into memory socket 0, and a 6116 RAM chip into memory socket 3. Set the 7804 card links as specified and insert it into an STD card frame. Set the 7304 card links as described in section 2.4, and insert it into the STD card frame. Connect the terminal, turn on the power, press pushbutton reset, and the monitor should display a sign-on prompt on the VDU terminal. No other initialisation is required.

2.4 Prolog 7880 Z80 Processor Card

The 7880 card requires a separate dual UART card to support the BWS-1000 Z80 monitor program. The standard monitor version (BWS-1003) assumes that the Prolog 7304 dual UART card is installed in the system. Setup requirements for this UART card are discussed in section 2.5.

Links on the 7880 card must be set up to the factory installation as per the Prolog manual, to provide:

- . Memory socket 0 address mapped to 0000 to 0FFF Hex as 2732 type EPROM.
- . Memory socket 1 address mapped to 3000 to 37FF Hex as 6116 type RAM.

2.0 Hardware Requirements and Configuration (contd)

To install the BWS-1000 Z80 monitor program, insert EPROM BWS-1003 into memory socket 0, and a 6116 RAM chip into memory socket 1. Set the 7880 card links as specified and insert it into an STD card frame. Set the 7304 card links as described in section 2.4, and insert it into the STD card frame. Connect the terminal, turn on the power, press pushbutton reset, and the monitor should display a sign-on prompt on the VDU terminal. No other initialisation is required.

2.5 Prolog 7304 Dual UART Card

The Prolog 7304 dual UART card provides the serial ports required for the &TITLE2 Z80 monitor program to work with any of the Prolog 7303, 7304, 7380 processor cards.

Links on the 7304 card must be set up to the factory installation as per the Prolog manual, to provide:

- . Card addressing at I/O location E0 hex.
- . Standard baud rate selection.
- . Channel A configured as DCE, channel B configured as DTE.

The terminal VDU is connected to channel A, which is initialised to 9600 baud, 8 data bits, no parity bit, 1 stop bit, no interrupts. CTS and DSR are asserted.

RS-232C connections for channel A are as follows:

pin 1	protective ground
pin 2	data received by 7304 card
pin 3	data transmitted from 7304 card
pin 4	RTS sensed by 7304 card (not used by monitor)
pin 5	CTS asserted by 7304 card
pin 6	DSR asserted by data terminal
pin 7	signal ground
pin 20	DTR sensed by 7304 card (not used by monitor)

Channel B is initialised to 2400 baud, 8 data bits, no parity bits, 1 stop bit, no interrupts. DTR and RTS are asserted.

RS-232C connections for port 2 are as follows:

pin 1	protective ground
pin 2	data transmitted from 7304 card
pin 3	data received from 7304 card
pin 4	RTS asserted by 7304 card
pin 5	CTS sensed by 7304 card (not used by monitor)
pin 6	DSR sensed by 7304 card (not used by monitor)
pin 7	signal ground
pin 20	DTR asserted by 7304 card

3.0 COMMAND SUMMARY AND SYNTAX

3.1 Command Summary

General:

?	Display a command summary
H	Hexadecimal addition/subtraction calculator
C	Jump to Command Basic (if installed)
T	Talk to host CPU (terminal mode)

Memory Examination/Modification:

D	Display memory block in Hex and ASCII
M	Memory byte examine/alter
W	Memory word examine/alter/trace
F	Fill a memory block with a constant
K	Copy a memory block
V	Verify (compare) two memory blocks
Q	Test a memory block

Register Examination/Modification:

R	Register, byte wide, examine/alter
P	Register pair, word wide, examine/alter/trace
X	Display complete saved CPU state

I/O Data Transfer:

O	Output data via I/O port
I	Input data via I/O port

Program Execution Control:

G	Go to target execution address
S	Single step the target code (only with Hardware Debug card)
B	Set/examine/remove software breakpoints
A	Set/examine/remove hardware address compare breakpoint (only with Hardware Debug card)
Z	Set/alter breakpoint display flag

INTEL-HEX File Transfer:

L	Load a memory image from an INTEL-HEX format file received via the auxiliary serial port
U	Upload a memory image as an INTEL-HEX format file via the auxiliary serial port

3.0 Command Summary and Syntax (contd)

3.2 Command Syntax

BWS-1000 monitor commands consist of a one letter (case irrelevant) command character, and are generally followed by one or more hex numeric parameters appropriate to the command function. These numeric parameters may be separated by either a comma or a space. Command lines are interpreted by the monitor when terminated with a Carriage Return (CR). A backspace character may be used to 'rubout' erroneous characters in a command line before the CR is entered.

When monitor commands cause a long display to be sent to the terminal screen (such as a long memory dump), the display can be paused, or frozen, at any time by typing any key (except CR which causes a return to prompt level). Once frozen, any other key will resume the display output to the screen.

Invalid commands will cause the prompt "Command Error !". Re-enter the correct command on a new line. If the BWS-2000 Hardware Debug card is not installed, commands which require this card will not execute, but will generate the Command Error prompt.

If the BWS-2000 Hardware Debug card is installed, the monitor can be "warm booted" at any time by pressing the push button on the debug card. This warm boot removes any breakpoint instructions that were inserted into the target code, but does not re-initialise breakpoints, register values, etc that have been set within the monitor.

The monitor may be "cold booted" at any time by entering a "control R" character. In addition, the monitor will execute a "JMP 0000" instruction at any time by entering a "control C" character. This function performs a cold boot for ROM based monitor versions, and a return to CP/M for the CP/M based monitor.

3.0 Command Summary and Syntax (contd)

3.3 Symbol Definitions

aaaa - Hexadecimal 16 bit address
 dd - Hexadecimal 8 bit data
 dddd - Hexadecimal 16 bit data
 char - ASCII character
 n - Decimal count
 start,end,new - Hexadecimal memory addresses
 port - 8 bit I/O port address

 r - Single register names;
 A (a) A' (a') Main & alternate accumulators
 F (f) F' (f') Main & alternate flags (SZ_H_VNC)
 B (b) B' (b')
 C (c) C' (c')
 D (d) D' (d') Main & alternate
 E (e) E' (e') 8 bit general purpose
 H (h) H' (h') registers
 L (l) L' (l')

 p - Register pairs;
 AF (a) A'F' (a') Main & alt. accumulator and flags
 BC (b) B'C' (b') Main & alternate
 DE (d) D'E' (d') 16 bit general purpose
 HL (h) H'L' (h') register pairs

 IX (x) Index register IX
 IY (y) Index register IY

 SP (s) Stack pointer
 PC (p) Program counter
 INT (i) Interrupt Vector and Status

4.0 COMMAND DESCRIPTIONS

4.1 Help Query

Syntax: ?

Parameters: none

Display a summary of all BWS-1000 commands (except '?').

NOTE: The command summary display includes all BWS-1000 monitor functions, including the additional functions that require the BWS-2000 Hardware Debug card to be installed. However, if these additional command functions are invoked without the Hardware Debug card in the system, the monitor will return a Command Error.

?????
? ?
?
??
??

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

	A
	AA AA
	A A
	AAAAAAA
4.2 Set/Examine/Remove Hardware Address Compare	A A
-----	A A

WARNING: This function requires the BWS-2000 Hardware Debug card. Invoking this function without the debug card will cause a Command Error.

Syntax: A
 Aaaaa
 Aaaaa,n

Parameters: aaaa = hardware address
 n = matching loop count (0 means clear)

'A' - Display the current address compare location in the format:

address n count

where:

address = address of breakpoint
 n = pass count (number of address matches before
 a return to monitor prompt level)
 count = number of address matches that have occurred

'Aaaaa' - Set the hardware address compare facility of the BWS-2000 Hardware Debug card to the specified address. When an instruction is executed (M1 cycle) with matching address on the bus an NMI is generated and the monitor is re-entered. The message "Address match at dddd $\frac{1}{2}$ Break = aaaa" and the appropriate CPU status as determined by the breakpoint "Z" flag are printed, and the monitor returns to prompt level.

'Aaaaa,n' - Set the hardware address compare as above, and loop "n" address matches before returning to monitor prompt level. The CPU status will be printed at each loop and/or at the final monitor re-entry as determined by the breakpoint "Z" flag. The command can be terminated before the "n" passes have been completed by typing a CR, which is detected at the next monitor re-entry.

'Aaaaa,0' - Cancel the address comparison at the specified location. (Actually the compare address is set to FFFF, which is UNLIKELY to be accessed by a normal program).

4.0 Command Descriptions (contd)

BBBBB
 B B
 BBBBB
 B B
 BBBBB

4.3 Set/Examine/Remove Software Breakpoints

Syntax: B
 Baaaa
 Baaaa,n

Parameters: aaaa = breakpoint address
 n = breakpoint loop count (0 means clear)

'B' - Display the active software breakpoints in the format:

address n count

where:

address = address of breakpoint
 n = pass count (number of passes through the
 breakpoint before a return to monitor
 prompt level)
 count = number of breakpoint passes that have occurred

'Baaaa' - Set up a software breakpoint at the specified address. When the user program is (re)started using the 'G' command, a RST 3 instruction (opcode= DF hex, vector @18 hex) temporarily replaces the instruction in the target code, and thus causes a monitor re-entry on execution of the instruction. The message "Break = aaaa" and the appropriate CPU status is printed, and the monitor returns to the prompt level.

'Baaaa,n' - Set up a software breakpoint as above, and loop "n" address matches before returning to monitor prompt level. The CPU status will be printed at each loop and/or at the final monitor re-entry as determined by the breakpoint "Z" flag. The command can be terminated before the "n" passes have been completed by typing a CR, which is detected at the next monitor re-entry.

'Baaaa,0' - Clear the software breakpoint at the specified address

4.0 Command Descriptions (contd)

4.3 Set/Examine/Remove Breakpoints (contd)

NOTES ON BREAKPOINTS

- . Software breakpoints can only be inserted into RAM based user program. Use the hardware address compare facility or the single step facility of the BWS-2000 Hardware Debug card if a breakpoint or a trace through ROM based code is desired.
- . The breakpoint table is currently 16 deep, allowing 16 breakpoints to be active simultaneously. A diagnostic is displayed if the user attempts to overflow this table.
- . Breakpoints which may be multiply passed during user program execution must be treated with some caution without the BWS-2000 Hardware Debug card. When the monitor restarts a user program after a software breakpoint is encountered, the original instruction is replaced into the user program for execution. If the Hardware Debug card is installed, the monitor single steps over this instruction, and then replaces it again with the software breakpoint RST 3 instruction to allow for multiple passes through the breakpoint. If the Hardware Debug card is NOT installed, the breakpoint instruction will not be re-inserted into the user code until the next monitor re-entry occurs, and thus multiple pass breakpoints may not occur.

The solution when the Hardware Debug card is NOT installed is to insert TWO or more software breakpoints into a loop where multiple loop breakpoints are desired. As each breakpoint is encountered, the other breakpoint instructions will be replaced into the user program as the monitor restarts the program after the breakpoint.

4.0 Command Descriptions (contd)

4.3 Set/Examine/Remove Breakpoints (contd)

Example: Consider a program loop between 1000h and 1100h. To successfully break at each pass of the loop at say address 1003h (with the next instruction at address 1005h), insert breakpoints as follows:-

B1003
B1005

At each pass of the loop;

a software breakpoint will occur at address 1003h, the software breakpoint at address 1005h will be inserted, the original instruction at address 1003h will be replaced and executed;

a software breakpoint will occur at address 1005h, the software breakpoint at address 1003h will be inserted, the original instruction at address 1005h will be replaced and executed;

etc.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

CCCC

C C

C

C

4.4 Call Command Basic (if installed)

C C

CCCC

Syntax: C

Parameters: none

Call ByteWyde Systems BWS-1100 Command Basic, if this language is installed with the BWS-1000 monitor program.

Note: This command will cold boot the monitor if Command Basic is not installed.

NOTE: This function is not yet implemented

4.0 Command Descriptions (contd)

```

DDDDDD
D      D
D      D
D      D
D      D
DDDDDD

```

4.5 Display a Memory Block

Syntax: D
 Dstart
 Dstart,end

Parameters: start = start of memory dump
 end = end of memory dump
 (default = start + 100h)

Display the specified memory block in hex and ASCII. If end is not specified then it defaults to start + 100 hex. If start is not specified, then it defaults to the previous end value.

The display can be temporarily halted by typing any character other than CR, and restarted by typing any character. The display can be terminated at any time by typing CR.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

FFFFFFF

F

FFFF

F

F

4.6 Fill a Memory Block

Syntax: Fstart,end,dd

Parameters: start = start of memory dump
end = end of memory dump
dd = data to put into memory

The memory block:

start --> end

is filled with the specified value (dd).

4.0 Command Descriptions (contd)

GGGGG
G G
G
G GG
G G
GGGGG

4.7 Go to Target Execution Address

Syntax: G
Gaaaa

Parameters: aaaa = target execution address

The CPU state is totally restored from the saved state and execution of the target code is resumed. If an address is specified, this value is loaded into the program counter (PC) before transfer to the target code. If any software breakpoints have been set up (using the B command), the appropriate target code instructions are replaced with RST 3 instruction before target code execution commences.

NOTE: The current loop count for software breakpoints, hardware address match and single step are reset to zero before execution transfers to the target code.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

H H
H H
HHHHHHH
H H
H H

4.8 Hexadecimal Calculator

Syntax: Hexpression

Parameters: expression = hexadecimal expression to
evaluate

The hexadecimal expression is evaluated, and the result displayed in hexadecimal format. The only legal operators accepted by this command are '+' and '-'.

This function can be used to determine the two's complement of a number by using the command:

H-dddd

4.0 Command Descriptions (contd)

IIIII

I

I

I

I

4.9 Input Data via I/O Port

IIIII

Syntax: Iport

Parameters: port = 8 bit I/O port

The data input from the specified port is displayed as a two digit hexadecimal number.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

K K
K K
KK
K K
K K
K K

4.10 Copy a Memory Block

Syntax: Kstart,end,new

Parameters: start = start of old memory block
 end = end of old memory block
 new = start of new memory block

The memory block:

start --> end

is copied into the block:

new --> new + (start-end)

byte by byte in ascending address order using an LDIR instruction.

4.0 Command Descriptions (contd)

L
L
L
L
L
L
LLLLLL

4.11 Load an INTEL-HEX Format File

Syntax: L
Ldddd

Parameters: dddd = load offset constant

An INTEL-HEX format file is loaded into memory via the auxiliary serial port. If an offset is specified this modifies the default load address specified in the hex records.

Before commencing the load, the monitor will pass on a single command line, entered from the terminal, through the auxiliary serial port. In effect, the monitor goes into terminal mode for one line entry. This command line can be used to trigger the transmission of the INTEL-HEX file from the host CPU. Note that the host CPU is responsible for echoing this message; no local echo is provided by the monitor. The prompt "|" indicates that the monitor is ready to pass on the single command line through the auxiliary serial port.

The single command line must be terminated with a CR, which is also passed up through the auxiliary serial port, and the monitor then expects the downline load to commence immediately.

The hex file may be simultaneously displayed on the console as it is loaded. This display may be toggled on/off by typing control V.

If a checksum error is detected, the load will abort with an error message.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

MM MM
M M M M
M M M
M M
M M

4.12 Examine/Alter Memory Byte

Syntax: Maaaa

Parameters: aaaa = address to examine/alter

The address and current contents are displayed in the format:

aaaa dd _

User responses may be:

CR	move to the next byte location
-	move to the previous byte location
dd	replace the current contents with the new hex value (2 digits long), then move to the next byte location
'char	replace the current contents with the ASCII value of the entered character, then move to the next byte location (upper case only).
other	return to prompt

4.0 Command Descriptions (contd)

000
0 0
0 0
0 0
0 0
000

4.13 Output Data via I/O Port

Syntax: Oport,dd

Parameters: port = 8 bit I/O port address
dd = 8 bit data to be output

The 8 bit data value is output to the specified port.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

PPPPPP
P P
PPPPPP
P
P

4.14 Examine/Alter/Trace Register Pair

Syntax: Pp
Pp'

Parameters: p = register pair selection
p' = alternate register pair selection

where p = a	AF	and p' = a'	AF'
b	BC	b'	BC'
d	DE	d'	DE'
h	HL	h'	HL'
x	IX		
y	IY		
s	SP		
p	PC		
i	INT		

The current register pair's name and contents are displayed in the format:

p dddd _

User responses may be:

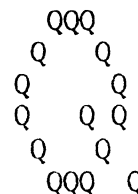
CR	move to next register pair
@	change to display of memory word (W command) using the contents of the displayed register pair as the current address
dddd	replaces the contents of the displayed register pair with the new 16 bit hex value, then moves to the next register pair
other	returns to prompt level

NOTES: The 16 bit hex numbers must be terminated by a carriage return. Less than 4 digits are right justified, with unentered digits set to zero. If more than 4 digits are entered then only the 4 rightmost digits are accepted.

The Interrupt register pair comprises the Z80 interrupt vector register in the high byte. Bit 2 of the low byte determines whether interrupts are enabled or disabled on entry to a user program, i.e.

Bit 2 = 0	- Disable Interrupts on program entry
Bit 2 = 1	- Enable Interrupts on program entry

4.0 Command Descriptions (contd)



4.15 Test a Memory Block

Syntax: Qstart,end

Parameters: start = start of memory block to be tested
end = end of memory block to be tested

The memory block:

start --> end

is tested by a simple write/readback test with a test number incremented at each memory location and at the end of each pass through the test block of memory. After each complete pass through the memory block a '+' is displayed.

The message:

address dd should = nn

is displayed for each location in error.

Type carriage return to abandon the test and return to monitor prompt level.

NOTE: Memory is tested in blocks of 100 hex. The test address range should therefore be multiples of 100 hex for the test to work correctly.

4.0 Command Descriptions (contd)

```

RRRRRR
R      R
RRRRRR
R      R
R      R

```

4.16 Examine/Alter Single Register

Syntax: Rr
Rr'

Parameters: r = single register selection
r' = alternate single register selection

where:	r =	a	A	r' =	a'	A'
		f	F		f'	F'
		b	B		b'	B'
		c	C		c'	C'
		d	D		d'	D'
		e	E		e'	E'
		h	H		h'	H'
		l	L		l'	L'

flags (F F') contain:	S	sign
	Z	zero
	H	half carry
in order: SZ_H_VNC	V	overflow/parity
	N	add/subtract
	C	carry

The current register's name and contents are displayed in the format:

r dd _

User responses may be

CR	move to the next byte register
dd	replaces the current contents of the byte register with the new 8 bit data, then moves to the next byte register.
other	return to prompt level

4.0 Command Descriptions (contd)

SSSS
S S
SSS
SSS
S S
SSS

4.17 Single Step Target Code

WARNING: This function requires the BWS-2000 Hardware Debug card. Invoking this function without the debug card will cause a Command Error.

Syntax: S
Sn

Parameters: n = number of instructions to step

Single step the target code by one (or n) instructions, then return to monitor prompt level.

This facility utilizes the hardware count down facility of the Hardware Debug card to generate an NMI during execution of the next instruction of the target code. Stepping over n instructions is accomplished within the monitor by re-entering the target code until the required number of target instructions have been executed.

NOTE: The current loop count for software breakpoints, hardware address match and single step are reset to zero before execution transfers to the target code.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

TTTTTTT

T
T
T
T

4.18 Talk to host CPU

Syntax: T

Parameters: none

All subsequent characters entered at the system terminal are passed directly on through the auxiliary serial port. All characters received through the auxiliary serial port are displayed on the system terminal. No intermediate character processing occurs.

Characters entered from the system terminal in this mode are passed onto the auxiliary serial port as 7 bit ASCII characters. Transmission from the serial port is controlled by the parity setting of the auxiliary serial port (initialised to 8 data bits, no parity).

The terminal mode may be terminated by entering "cntrl E" at the system terminal, whereby the monitor returns to its normal mode.

NOTE: When entering terminal mode, the monitor does NOT flush a character which may have been received at the auxiliary serial port. Such a character will be displayed as the first character received when the terminal mode is entered.

4.0 Command Descriptions (contd)

U U
U U
U U
U U

4.19 Upload an INTEL-HEX Format File

UUU

Syntax: Ustart,end

Parameters: start = start of memory block to be
transmitted
end = end of memory block to be
transmitted

The memory block:

start --> end

is transmitted in INTEL-HEX format via the auxiliary serial port.

The hex file may be simultaneously displayed on the console as it is transmitted. This console display may be toggled on/off by typing control V. The upload procedure may also be aborted at any time by typing control R.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

V V V
V V V
VV
V

4.20 Compare (Verify) Two Memory Blocks

Syntax: Vstart,end,new

Parameters: start = start of first memory block
end = end of first memory block
new = start of second memory block

The blocks of memory:

start --> end
new --> new + (start-end)

are compared. The error message;

address dd should = nn

is displayed for each mismatched byte.

4.0 Command Descriptions (contd)

```

W      W
W  W  W
W  W W W
W W  W W
WW    WW

```

4.21 Examine/Alter/Trace Memory Word

Syntax: Waaaa

Parameters: aaaa = address of word to be examined

The current address and current contents are displayed in the format:

```

aaaa      dddd      _

```

User responses may be:

```

CR          move to next word location
-           move to previous word location
@           current address is set to the currently displayed
            location contents.
dddd        replaces current contents of location, then moves
            to next word location.
other       returns to prompt level

```

NOTE: The 16 bit hex numbers must be terminated by a carriage return. Less than 4 digits are right justified, with unentered digits set to zero. If more than 4 digits are entered then only the 4 rightmost digits are accepted.

BWS-1000 Z80 Monitor Program

4.0 Command Descriptions (contd)

```

X      X
X      X
      XX
X      X
X      X

```

4.22 Displayed Saved CPU State

Syntax: X

Parameters: none

Display the complete status of the CPU, that is the status saved away upon monitor re-entry and subsequently modified by either the R or P commands.

The following are displayed:

- CPU registers:

Set flags (displayed symbolically)
Register pairs

for both the main and alternate register banks.

- Current value of the I register.

- The interrupt status of the target code.

The display format is as follows:

	AF	BC	DE	HL	IX	IY	SP	PC	OP
SZ_H_VNC	dddd	dddd	dddd	dddd	dddd	dddd	dddd	dddd	dd
SZ_H_VNC	dddd	dddd	dddd	dddd	dd E/D				
± 7					/	±			
flags					I register	Interrupt status			

NOTE: OP is the next opcode to be executed at the location specified by PC

4.0 Command Descriptions (contd)

ZZZZZZZZ

Z

Z

ZZ

4.23 Set/Alter Breakpoint Display Flag

Z

ZZZZZZZZ

Syntax: Z
Zdd

Parameters: dd = flag status byte

'Z' - Display the current value of the breakpoint display flag
in the format:

dd

'Zdd' - Set the breakpoint display flag according to the status
code dd. This flag controls the response of the monitor at
re-entry. Only the two least significant bits of the flag
byte have any relevance, as follows:

bit 0 SET - display the CPU status (X command) on each
monitor re-entry, EXCEPT the terminating re-
entry, that is, the one that causes a
return to prompt level.

bit 1 SET - display the CPU status (X command) upon
return to monitor prompt level.

The most useful values for the flag are therefore:

- 0 no display of CPU status at all
- 2 display CPU status only upon return to prompt
level
- 3 display CPU status upon each re-entry

5.0 MONITOR ENTRY POINTS

5.1 Monitor Subroutines

The BWS-1000 monitor uses a number of general purpose subroutines for initialisation, console I/O and serial port I/O. These subroutines can be accessed from user programs by simply CALLing the routines at the monitor subroutine jump table.

The subroutines and their jump address are as follows:

Address (hex)	Label	Subroutine Function
0080	.cold	Monitor cold boot (initialisation)
0083	.warm	Monitor warm boot
0086	.const	Console status test (A=0 if ready, A=FF if not)
0089	.conin	Console data input (data in A)
008C	.conout	Console data output (data in C)
008F	.coni8	8 bit console data input (data in A)
0092	.remst	Auxiliary port status test (A=0 if ready, A=FF if not)
0095	.remin	Auxiliary port data input (data in A)
0098	.remout	Auxiliary port data output (data in C)
009B	.remi8	8 bit auxiliary port data input (data in A)
009E	.p2hx	Display A as 2 hex digits on console
00A1	.p4hx	Display HL as 4 hex digits on console
00A4	.pcrlf	Display CR/LF on console
00A7	.pmsg	Print (HL)-> string up to EOT
00AA	.pNext	Print string following call up to EOT
00AD	.g2hx	Get 2 hex digits into A from console buffer
00B0	.g4hx	Get 4 hex digits into HL from console buffer
00B3	.glin	Get new line from console
00B6	.talk	Talk through auxiliary port to remote host
00B9	.rhex	Load hex file from auxiliary port
00BC	.shex	Send (updump) hex file through auxiliary port
00BF	.passcl	Send one command line to remote host
00C2	.init	Initialise bios
00C5	.minit	Initialise Monitor RAM and bios

Example: CALL 0080 will cold boot the monitor

5.0 Monitor Entry Points (contd)

5.2 Interrupt Jump Vectors

The BWS-1000 monitor vectors all software and hardware restart jumps to the monitor RAM, to allow access from user programs. Four bytes are reserved in the monitor RAM for each restart vector, to allow a further jump instruction to be inserted as required.

The RST 3 and the NMI restarts are vectored back into the monitor at monitor initialisation. The other restart address are dummied with a RET instruction.

The restart RAM vectors are located at the start of monitor RAM. The absolute location of the vectors depends on the particular Prolog Z80 processor card in use. Relocatable locations are as follows. Consult the load map stapled to the back of this manual for the absolute address of the start of the RAM vector locations.

Relocatable Address	Label	Initialised To	Comments
	RAM_VEC		Start of Monitor RAM
0000	RST1_VEC	RET,FF,FF,FF	Jump from 08 hex
0004	RST2_VEC	RET,FF,FF,FF	Jump from 10 hex
0008	RST3_VEC	JP RST_ENT,FF	Jump from 18 hex (monitor software breakpoint)
000C	RST4_VEC	RET,FF,FF,FF	Jump from 20 hex
0010	RST5_VEC	RET,FF,FF,FF	Jump from 28 hex
0014	RST6_VEC	RET,FF,FF,FF	Jump from 30 hex
0018	RST7_VEC	RET,FF,FF,FF	Jump from 38 hex (maskable interrupt)
001C	NMI_VEC	JP NMI_ENT,FF	Jump from 66 hex (monitor restart, single step, address halt)
0020	MON_VEC	RET,FF,FF,FF	Command Table Extension
0024	M_RETURN	DI,RET	Return to Target Code

APPENDIX A - COMMAND SYNTAX SUMMARY

?	Display a command summary
Aaaaa,n	Set/examine/remove address compare
Baaaa,n	Set/examine/remove software breakpoints
C	Call Command Basic (if installed)
Dstart,end	Display memory block
Fstart,end,dd	Fill a memory block
Gaaaa	Go to target execution address
Hexpression	Hexadecimal calculator
Iport	Input data via i/o port
Kstart,end,new	Copy a memory block
L	Load an INTEL-HEX format file
Maaaa	Examine/alter memory byte
Oport	Output data via i/o port
Pp	Examine/alter/trace register pair
Qstart,end	Test a memory block
Rr	Examine/alter single register
Sn	Single step target code
T	Talk to host CPU
Ustart,end	Transmit an INTEL-HEX format file
Vstart,end,new	Compare (verify) two memory blocks
Waaaa	Examine/alter/trace memory word
X	Display saved CPU state
Zn	Set/alter breakpoint display flag
^C	Execute JMP 0000 instruction
^R	Cold boot the monitor
^V	Toggle the upload/load console display
^Z	Abort uploading or loading a hex file

0100 .START	4310 NMI_VE	4300 RST1_V	4304 RST2_V
4308 RST3_V	430C RST4_V	4310 RST5_V	4314 RST6_V
4318 RST7_V	0080 .VSTAR	1559 \$CONI8	1553 \$CONIN
1563 \$CONOU	1549 \$CONST	1599 \$INIT	157C \$REMIS
1576 \$REMIN	1587 \$REMOU	156D \$REMST	0080 .COLD
008F .CONI8	0089 .CONIN	008C .CONOU	0086 .CONST
00AD .G2HX	00B0 .G4HX	00B3 .GLIN	00C2 .INIT
00C5 .MINIT	009E .P2HX	00A1 .P4HX	00BF .PASSC
00A4 .PCRLF	00A7 .PMSG	00AA .PNEXT	009B .REMIS
0095 .REMIN	0098 .REMOU	0092 .REMST	00B9 .RHEX
00BC .SHEX	00B6 .TALK	0083 .WARM	12F2 @PASSC
1318 @RHEX	1380 @SHEX	12CA @TALK	14E3 CRLF
1453 GET2HX	1437 GET4HX	1472 GETLIN	1524 MON_CO
0800 MON_IN	080C MON_WA	14D0 PMSG	14CA PNEXT
14B5 PUT2HX	14A9 PUT4HS	011E .VCIN	0109 .VCINI
011B .VCINS	0100 .VCOLD	0121 .VCOST	0124 .VCOUT
010C .VIOIN	0118 .VKEYI	0115 .VKEYS	0106 .VMAIN
0112 .VVDU	010F .VVDUI	0103 .VWARM	01DB @MAIN
04DB GETCHR	04D3 GETSTA	01A4 INITC	0158 INITIO
0185 INITVD	0557 KEYIN	054F KEYSTA	051F PUTCHR
0513 PUTSTA	0203 VDU	0127 VDU_CO	013E VDU_WA
064E ENDTBL	0567 STRTBL	4210 VECTBL	0010 LCTCTB
420B ATTRIB	420C CHARSE	4200 CHRCNT	0334 CLRSCN
4235 COLTBL	01D0 COLWRT	4218 CTCTBL	0566 DEFATT
420A FLAG	015B INIOLP	01B7 INITCO	4204 INKEY
4229 IOTBL	4202 OUTPTR	4300 RAM_VE	420D TIMER
0026 VEC_LN	15AC VEC_ST	01F1 @LOCAL	4295 ESCVEC
4270 ESCTBL	42E7 CTLVEC	42DF CTLTBL	0327 BELLON
03F6 BRIGHT	022C CADDR	0345 CLRALL	0337 CLREOS
03E3 CLRLST	02EE CSDWN	0371 CSHOME	02DE CSLFT
0298 CSON	031A CSRET	02CA CSRG	4206 CSRSPD
0303 CSUP	4209 CURHOR	023E CURPOS	048F CURSOF
0485 CURSON	4208 CURVRT	03B6 DELLIN	03FE DIM
047E DISPOF	0477 DISPON	02AF DSWRT	0361 ERAEOL
0219 ESCCMD	0457 FLSHOF	044F FLSHON	131C GETREC
03CE INSLIN	040E INVOFF	0406 INVON	0399 MOVEDN
03EC NORMAL	049D RCVHEX	0455 SAVEAT	0380 SCROLL
045F SELCS0	0465 SELCS1	046B SELCS2	0471 SELCS3
04C2 SETATT	04B6 SRCHTB	041E ULNOFF	0416 ULNON
0426 VDIM	042F VDIM0	0433 VDIM1	0437 VDIM2
043B VDIM3	043F VDIM4	0443 VDIM5	0447 VDIM6
044B VDIM7	0283 VDU2	0244 XPOS	025C YPOS
4000 BUSBUF	4100 KEYBUF	04F5 CHARIN	0533 CHAROU
4203 KEYCNT	0560 KEYI7	4201 INFTR	4205 OUTKY
058C CORTBL	0636 CRTBL	063E CRVEC	056F CTRTBL
050B EIOTBL	0507 ERTBL	05EC ERVEC	0567 INTTBL
4210 PAVECT	4212 PBVECT	0580 PIOA	0585 PIOB
4226 TPS	0040 LINSIZ	1018 @BLOCK	104E @BREAK
0E76 @GALC	1003 @FILL	0F25 @FLAG	126A/@GOTO
0EEB @INCMD	0E97 @MEMDM	0F0D @OUTCM	0CD5 @PAIR
130F @RECEI	0DB8 @REGDU	0C19 @REGIS	137C @SEND
0FAC @TEST	1031 @VERCM	0F3C @VIEW	0F5D @WORD
1499 ASCHEX	14DA CRLFS	08BF GETHEX	4337 LINBUF
080F MON_PR	4479 MON_ST	4320 MON_VE	0C13 MS_FRM
0BC1 MS_SGN	4329 M_P1	432B M_P2	432D M_P3
4317 M_PAUS	0889 PARAMS	10EE BP_FIN	14EA ECHO
4417 M_PC	1505 OUTPUT	14AE PUT2HS	43FB REG_IM
1410 SPACE	0EB9 @DUMP	0FF8 MDATA	4335 M_LAST

43F7 M_ZFLA	0010 BP_LEN	0014 REG_LE	1072 @ADCOM
1072 @SINGL	4377 BP_TAB	0007 BP_MID	12C9 DEBINI
4324 M_RETU	43F5 M_SERV	43FA M_SSMO	116D NMI_EN
440F OLD_HL	4411 OLD_SF	1178 RST_EN	1468 HEXASC
4334 M_VFLA	4415 ZEND	4329 ZSTART	43E7 BP_AC
43EE BP_SS	4328 M_DBFL	43F8 M_HOPA	432F M_P4
4331 M_P5	2000 ENDCHA	1800 ROMCHA	